Computational Mathematics and AI

Lecture 1: Machine Learning Overview

Lars Ruthotto

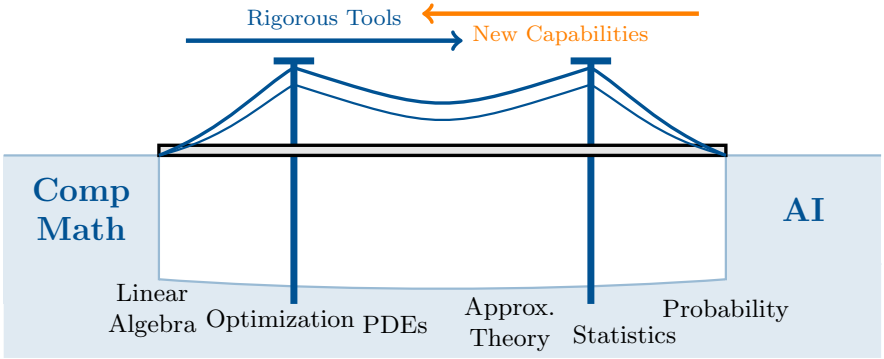
Departments of Mathematics and Computer Science

lruthotto@emory.edu

larsruthotto

slido.com #CBMS25

Overview: Computational Mathematics and AI



Questions or Feedback:

slido.com
#CBMS25

Module	Lectures	Theme
ML Crash Course	1-3	Architectures, optimization, generalization
ApplMath for ML	4-6	Theory, regularization, PDEs
ML for ApplMath	7-10	Operators, inverse problems, discovery

Reading List: Lecture 1

Historical Context: Machine learning builds on centuries of mathematical foundations: from Gauss's least squares (1809) to modern optimization theory.

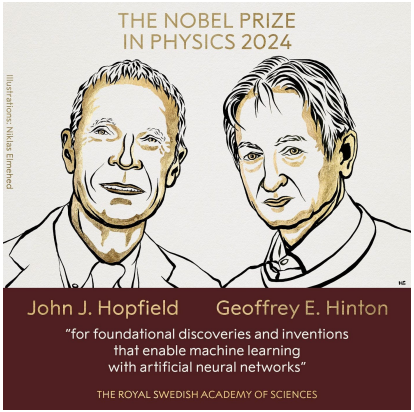
Key Readings:

1. Mitchell (1997) – *Machine Learning*, McGraw-Hill.
   Core ML definition and foundational concepts.
2. Wolpert and Macready (1997) – No Free Lunch Theorems.
   Fundamental limits of learning algorithms.
3. Higham and Higham (2019) – Deep Learning for Applied Mathematicians.
   Bridges applied math and deep learning.
4. Belkin (2021) – Fit without Fear.
   Discovery of the double descent phenomenon.
5. Ferguson et al. (2025) – The Future of AI and Mathematical & Physical Sciences.
   Workshop report highlighting mathematics role in AI.

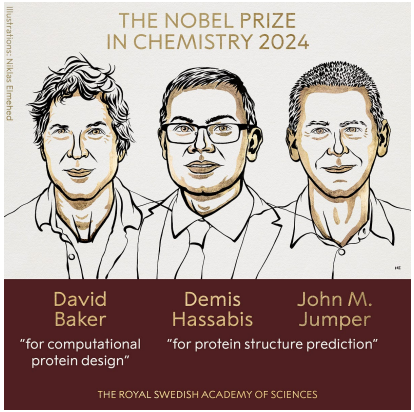
Lecture Outline: What is ML? $\rightarrow$ Learning Tasks $\rightarrow$ Bias-Variance & Double Descent $\rightarrow$ Course Overview

Motivation

Starting Point: 2024 Nobel Prizes



- Statistical mechanics + neuroscience
- Foundational methods for AI
- **Science** $\rightarrow$ **AI**



- Curated chemical & biological data
- AlphaFold: protein design & prediction
- **Science** $\leftrightarrow$ **AI**

Both built on **decades of basic research** in math and physical sciences

NSF Workshop on AI and Math & Physical Sciences

Workshop Methodology

- ≈60 invited experts across MPS domains
- Online surveys + in-person discussions
- Cross-cutting themes identification
- Topic-specific breakout sessions
- Draft report circulated for feedback

Key Discussion Areas

- AI capabilities driving scientific innovation
- Barriers to wider AI adoption in MPS
- Underutilized AI techniques in science
- New AI capabilities needed for discovery

Report available: Ferguson et al. 2025

Mathematics: The Foundational Backbone of AI

Optimization Theory

- Gradient descent and variants
- Stochastic optimization (SGD, Adam)
- Non-convex landscape analysis
- Convergence guarantees

Approximation Theory

- Universal approximation theorems
- Function spaces and norms
- Spectral methods
- Kernel methods and RKHS

Statistical Learning

- Generalization theory
- PAC-Bayes framework
- Empirical risk minimization
- Concentration inequalities

Linear Algebra

- Matrix factorizations
- Eigenvalue problems
- Randomized algorithms
- High-dimensional geometry

AI builds upon strong foundation in mathematics and statistics

Computational Mathematics Opportunities

Where computational math meets AI: A rich research landscape

Numerical Optimization

- Distributed optimization algorithms
- Non-convex optimization theory
- Adaptive learning rate schedules

Numerical Linear Algebra

- Randomized algorithms for ML
- Preconditioning strategies
- Low-rank approximations

Algorithm Discovery

- AI-discovered numerical schemes
- Adaptive discretizations
- DARPA DIAL program initiatives

Numerical PDEs & AI

- Physics-informed neural networks
- Neural operator learning
- Interpreting AI via PDEs

High-Dimensional Problems

- Monte Carlo + deep learning
- Stochastic differential equations
- Control problems in high dimensions

Uncertainty Quantification

- Bayesian computational methods
- Error analysis for AI models
- Generalization bounds

Course goal: Give a glimpse into these intersections

Lecture 1 Roadmap

1. What is Machine Learning?
   - ML as approximation science
   - Mathematical framework
2. Learning Tasks
   - Survey: Supervised, unsupervised, operator, RL, generative
3. Learning Theory
   - Classical: Bias-variance tradeoff
   - Demo: Polynomial fitting
   - Modern: Double descent phenomenon
4. Course Structure

What is Machine Learning?

Machine Learning: Core Definition

Mitchell (1997)

A program **learns** from experience E with respect to task T and performance measure P, if performance at T (measured by P) improves with experience E

ML as Approximation Science:

- Find functions that approximate relationships in data
- Balance approximation quality vs. generalization
- Trade-off between model complexity and data fit

Historical Connections:

- Gauss (1809): Least squares method
- Fisher (1922): Maximum likelihood estimation

ML = data-driven approximation theory

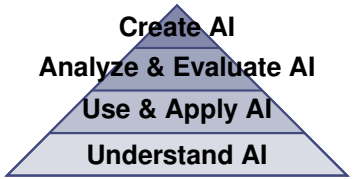
No Free Lunch Theorem

Wolpert (1997)

Without assumptions or prior knowledge, no learning algorithm generalizes better than any other (averaged over all problems)

Implications:

- Domain knowledge is *essential*
- Must match methods to problem characteristics
- Success requires exploiting problem structure



The Virtuous Cycle Exploits This:

- Computational math provides domain knowledge
- Scientific ML methods exploit structure
- Symmetries, conservation laws, multi-scale behavior

Generic methods fail; domain expertise wins; AI literacy is key

Mathematical Framework: Linear Regression

Setup: Data $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$ with $\mathbf{x}_i \in \mathbb{R}^d$, $\mathbf{y}_i \in \mathbb{R}$.

Model: $f_{\boldsymbol{\theta}}(\mathbf{x}) = \mathbf{x}^\top \boldsymbol{\theta}$ **Hypothesis class:** $\{f_{\boldsymbol{\theta}} : \boldsymbol{\theta} \in \mathbb{R}^d\}$ **Goal:** Find optimal $\boldsymbol{\theta}$

Optimization View

Minimize empirical squared error:

$$\min_{\boldsymbol{\theta}} \underbrace{\frac{1}{n}\|\mathbf{X}\boldsymbol{\theta} - \mathbf{y}\|_2^2}_{\text{empirical risk } \hat{L}(\boldsymbol{\theta})}$$

$\mathbf{X} \in \mathbb{R}^{n \times d}$ (rows = samples), $\mathbf{y} \in \mathbb{R}^n$

Statistical View

Minimize expected risk:

$$\min_{\boldsymbol{\theta}} \underbrace{\mathbb{E}_{(\mathbf{x},\mathbf{y})}[(f_{\boldsymbol{\theta}}(\mathbf{x}) - \mathbf{y})^2]}_{\text{expected risk } L(\boldsymbol{\theta})}$$

noise model $\mathbf{y} = \mathbf{x}^\top \boldsymbol{\theta}^* + \varepsilon$, $\varepsilon \sim \mathcal{N}(0, \sigma^2)$

Optimality Condition:

$$\hat{\boldsymbol{\theta}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y} \quad \text{(Normal Equations)}$$

Same optimality condition but different tools yield different results

Learning Tasks

Supervised Learning: Learn from Labeled Examples

Problem: Given $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$, find $F_{\boldsymbol{\theta}}(\mathbf{x}) \approx \mathbf{y}$

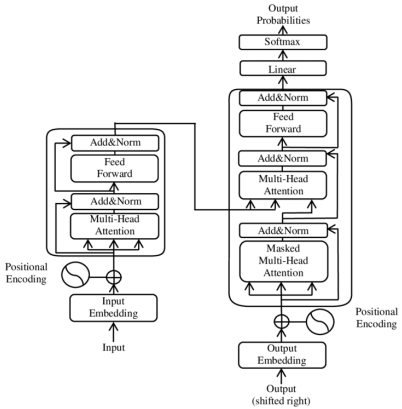
Example: Next Token Prediction (LLMs)

- Input $\mathbf{x}$: tokens "The cat sat on the"
- Output $\mathbf{y}$: next token "mat"
- Learn: distribution over vocabulary

Computational Math Connections:

- Function approximation theory
- Regularization & inverse problems
- Optimization algorithms

→ Lectures 2-3: Architectures & Learning



Godoy (CC BY 4.0)

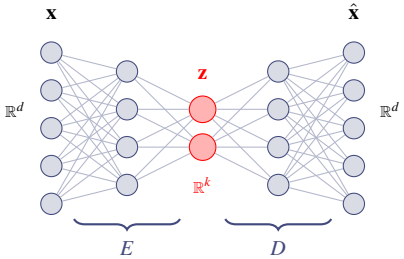
key insight: Most common ML task — foundation for classification & regression

Unsupervised Learning: Discover Hidden Structure

Problem: Given unlabeled $\{\mathbf{x}_i\}_{i=1}^n$, discover structure: $\mathbf{z} = E(\mathbf{x})$, $\hat{\mathbf{x}} = D(\mathbf{z}) \approx \mathbf{x}$

Example: Autoencoders

- Encoder E: $\mathbf{x} \in \mathbb{R}^d \to \mathbf{z} \in \mathbb{R}^k$ ($k \ll d$)
- Decoder D: reconstruct $\hat{\mathbf{x}}$ from $\mathbf{z}$
- Learn: compact latent representation



Computational Math Connections:

- Spectral methods (PCA = linear AE)
- Manifold learning & diff. geometry
- Dimensionality reduction for PDEs

key insight: Find low-dimensional structure without labels

Operator Learning: Maps Between Function Spaces

Problem: Learn $\mathcal{G} : \mathcal{U} \to \mathcal{V}$ between function spaces, e.g., $\mathcal{G} : \kappa(\cdot) \mapsto u(\cdot)$

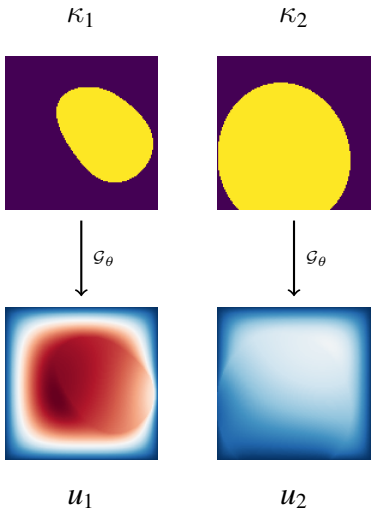
Example: Darcy Flow (Lecture 7)

- Input κ: permeability field
- Output u: pressure/solution field
- PDE: $-\nabla \cdot (\kappa \nabla u) = f$

Computational Math Connections:

- Green's functions & integral operators
- Many-query: UQ, inverse, control
- Surrogate modeling

→ Lectures 7: Scientific ML for PDEs



key insight: Amortize PDE solves — train once, evaluate many times

Reinforcement Learning: Learn from Interaction

Problem: Learn policy $\pi : \mathcal{S} \to \mathcal{A}$ maximizing reward $\mathbb{E}\left[\sum_{t=0}^{T} r(s_t, a_t)\right]$

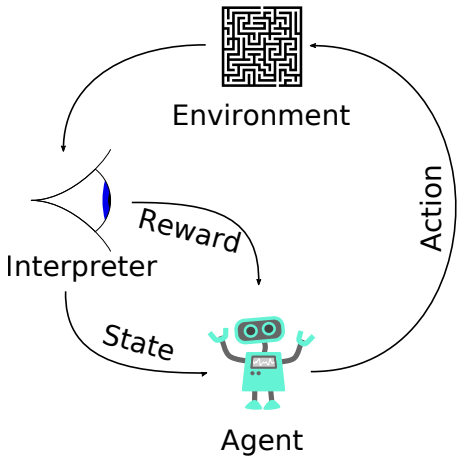
Example: Matrix Mult. (AlphaTensor)

- State s: tensor to decompose
- Action a: rank-1 elimination
- Result: 4×4 in 47 mults (was 49)

Computational Math Connections:

- Optimal control & dynamic programming
- Hamilton-Jacobi-Bellman equations
- Algorithm discovery (DARPA DIAL)

→ Lecture 8: High-Dim Optimal Control
→ Lecture 10: Math Discovery with AI



"Intelligence means having a goal" – Richard Sutton

Generative Modeling: Learn Probability Distributions

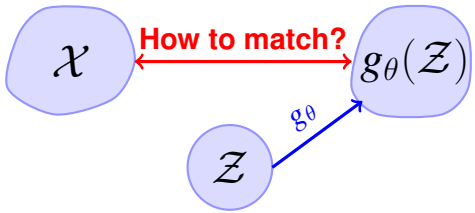
Problem: Given samples $\{\mathbf{x}_i\} \sim p_{\text{data}}$, learn $p_\theta \approx p_{\text{data}}$

Example: Posterior Distributions

- Observations $\mathbf{y}$, model $\mathbf{y} = F(\boldsymbol{\theta}) + \epsilon$
- Learn: posterior $p(\boldsymbol{\theta}|\mathbf{y})$
- Use: diffusion models as learned priors

Computational Math Connections:

- Optimal transport theory
- SDEs & Fokker-Planck equations
- Flow matching & continuity equations



→ Lecture 6: Generative Modeling via PDEs
→ Lecture 9: Bayesian Inverse Problems

Learning complex probabilit distributions: useful foundation for Bayesian inference

A Glimpse into Learning Theory

Classical Learning Theory: Capacity and Generalization

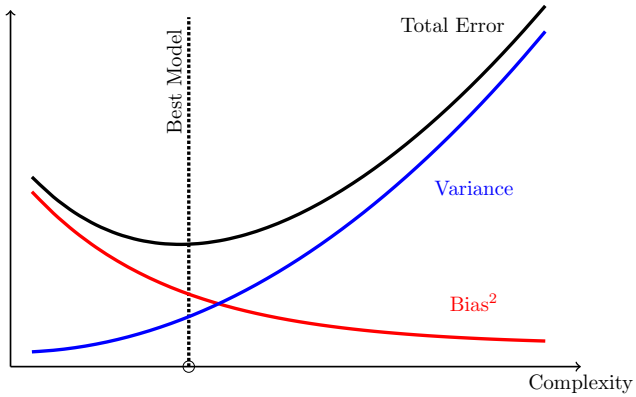
Model Capacity: How complex can a hypothesis class be?

VC Dimension: Maximum n points that can be shattered

- Linear classifiers in $\mathbb{R}^d$: VC dim $= d + 1$
- Polynomials of degree p: VC dim $= p + 1$

Bias-Variance Decomposition:

$$\underbrace{\mathbb{E}[(\hat{f} - y)^2]}_{\text{Test Error}} = \underbrace{\text{Bias}^2}_{\text{underfit}} + \underbrace{\text{Var}}_{\text{overfit}} + \sigma^2$$

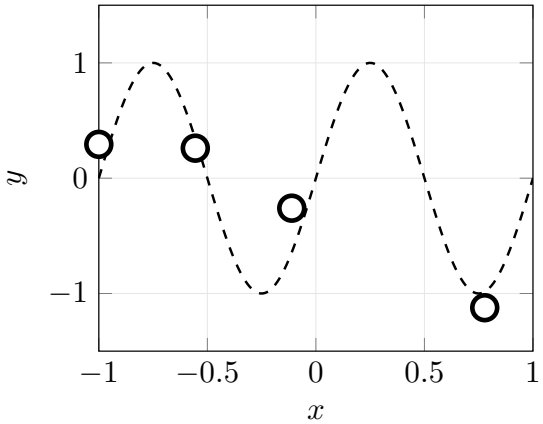


- **Bias:** Error from model assumptions
- **Variance:** Sensitivity to training data

Classical advice: Choose capacity to balance bias and variance

Classical Learning Theory: Polynomial Fitting

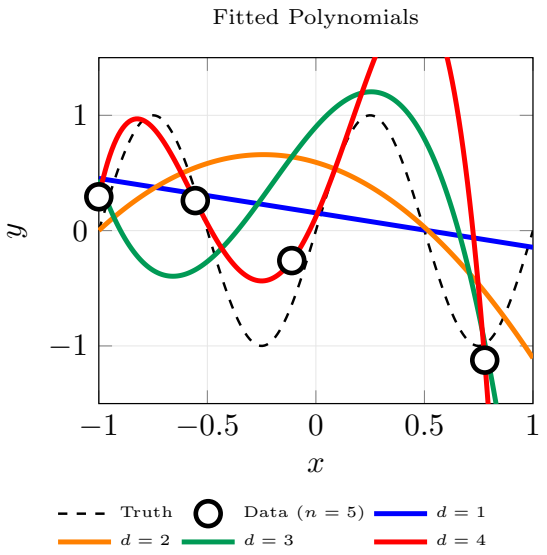
Setup: Fit Legendre polynomials to 5 noisy data points from $f(x) = \sin(2\pi x)$

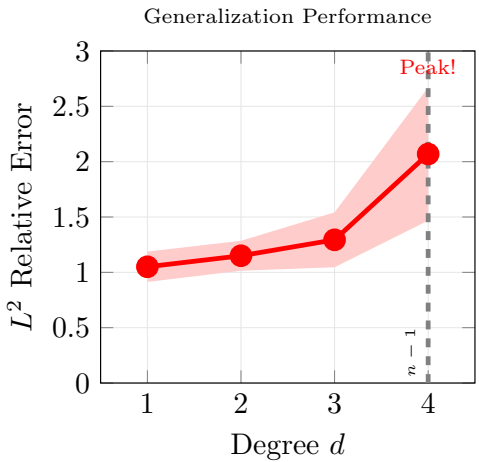


Question: How does polynomial degree affect fit quality?

Classical Learning Theory: Polynomial Fitting

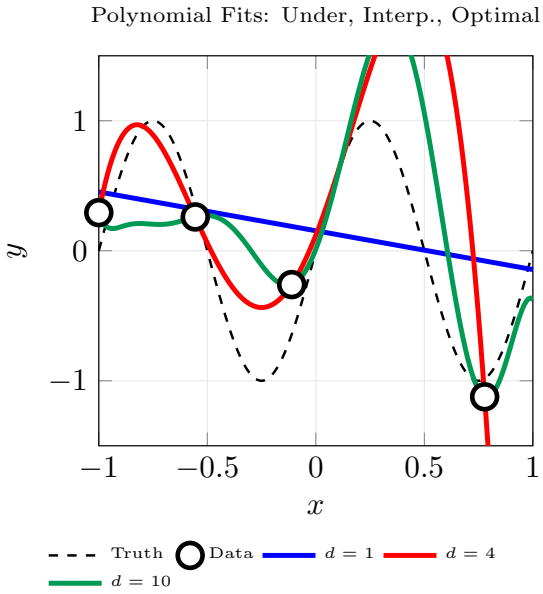
Setup: Fit Legendre polynomials to 5 noisy data points from $f(x) = \sin(2\pi x)$

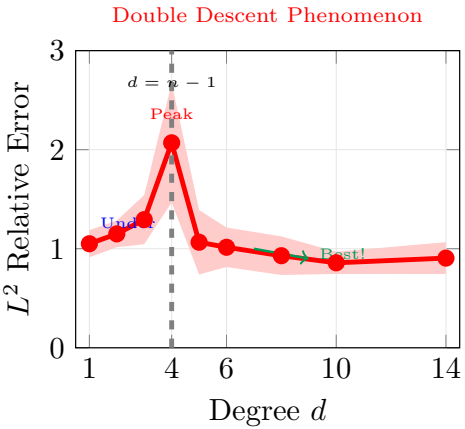




Classical wisdom: pick $d < 4$ to avoid overfitting and oscillations

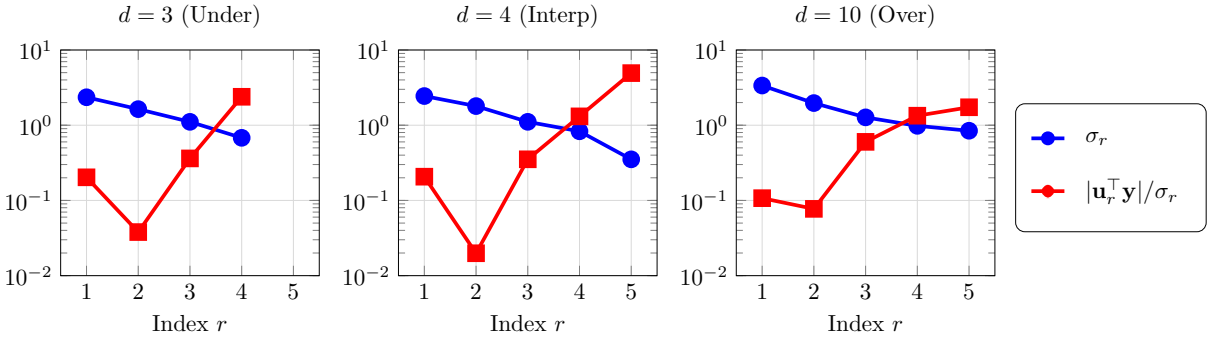
Double Descent: The Overparameterized Regime





Result: Overparameterization improves generalization via implicit regularization

Why Does Double Descent Happen?



Underparameterized

$$\boldsymbol{\theta} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$$

Threshold

$$\boldsymbol{\theta} = \mathbf{X}^{-1} \mathbf{y}$$

Overparameterized

$$\boldsymbol{\theta} = \mathbf{X}^\top (\mathbf{X}\mathbf{X}^\top)^{-1} \mathbf{y}$$

At threshold: small σ_r values amplify noise in poorly-sampled directions

Minimum norm bias in overparameterized regime acts as implicit regularization

Regularization: The Practical Solution

Explicit Regularization

Ridge Regression (L2):

$$\min_{\boldsymbol{\theta}} \|\mathbf{X}\boldsymbol{\theta} - \mathbf{y}\|_2^2 + \lambda\|\boldsymbol{\theta}\|_2^2$$

Solution:

$$\hat{\boldsymbol{\theta}} = (\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{y}$$

- $\lambda\mathbf{I}$ reduces influence of small σ
- established techniques to tune λ

Implicit Regularization

Gradient Descent:

- Zero initialization
- Iterative updates
- Converges to minimum norm solution

Overparameterized ($P \gg N$):

$$\hat{\boldsymbol{\theta}} = \mathbf{X}^\top(\mathbf{X}\mathbf{X}^\top)^{-1}\mathbf{y}$$

Algorithm chooses simpler solution

Need to understand regularization to understand generalization

Connection to Computational Mathematics

Ridge Regression = Tikhonov Regularization = Weight Decay

- Familiar tool for ill-posed inverse problems
- Stabilizes inversion of near-singular matrices
- Same mathematical principle in ML and comp math!

Modern Deep Learning:

- Relies on over-parameterization and *implicit* regularization
- SGD noise acts as regularizer
- Architecture choices matter (inductive bias)
- No explicit λ parameter needed (but other hyperparameters)

Forward Connections:

- Lecture 4: Implicit regularization of SGD
- Lecture 5: Advanced optimization methods

Computational mathematics provides tools to understand modern ML

Modern Phenomena Beyond Double Descent

Scaling Laws:

- Empirical power law relationships
- Performance vs. model size, dataset size, compute
- Example: GPT scaling laws, Chinchilla scaling
- *Test-time compute scaling:* More inference compute $\rightarrow$ better performance

Implicit Regularization:

- Gradient descent finds ”good” solutions without explicit penalties
- Zero initialization + GD $\rightarrow$ minimum norm solution
- SGD noise acts as implicit regularizer

These phenomena reappear throughout the course

- Developing mathematical tools for rigorous understanding

Modern ML challenges classical intuition

Course Overview and Summary

Course Structure: 10 Lectures, 3 Modules

Module 1: Crash Course

L1: ML Overview
- Learning tasks
- Double descent

L2: Learning Problems
- MLPs, GNNs, Transformers
- ResNets, Neural ODEs
- Loss functions

L3: Optimization
- Empirical vs. expected risk

Module 2: CM $\rightarrow$ AI

L4: Stochastic Optimization
- Convergence
- Implicit regularization

L5: Loss Landscapes
- Adaptive methods
- Modern optimization

L6: Generative Modeling
- PDEs, optimal transport
- Diffusion, flow matching

Module 3: CM $\leftarrow$ AI

L7: Scientific ML
- PINNs, neural operators
- learned solvers

L8: High-Dim PDEs
- Curse of dimensionality
- Deep BSDE, FBSDE, HJB

L9: Inverse Problems
- Simulation based inference
- Diffusion priors

L10: Math Discovery
- Evolutionary coding
- Proof assistants

Course Philosophy and Expectations

What this course IS:

- Illustrative: Representative examples from different topics
- Bidirectional: CompMath $\leftrightarrow$ AI synergy
- Hands-on: Numerical experiments and computational demos
- Research-oriented: Active frontiers, open problems

What this course is NOT:

- Comprehensive: 10 lectures cannot cover everything
- Pure theory: Balance rigor with intuition
- Software engineering: Concepts over production code
- Latest & greatest: Field evolves faster than curricula

Our approach:

- Pick characteristic issues from each research direction
- Guide you into the field, not exhaustive coverage
- Complement with workshop research talks
- Equip you to read papers and start your own projects

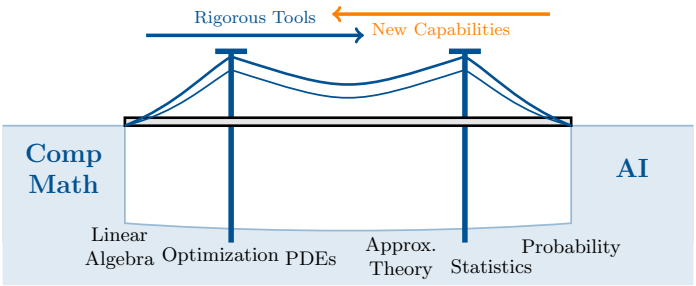
goal: Mathematical foundations + computational tools for CM+AI research

Cross-Cutting Themes

Watch for these recurring themes:

- **Data Efficiency:** Methods for limited, structured datasets
  - Manifold learning, sparse recovery, nonlinear approximation
- **Uncertainty Quantification:** Characterizing prediction confidence
  - Bayesian approaches, Monte Carlo, polynomial chaos
- **Multi-Scale Simulations:** Bridging temporal/spatial scales
  - Homogenization, multigrid, closure models
- **Physics-Informed Methods:** Combining ML with mechanistic models
  - PINNs, neural ODEs, differentiable physics
- **Curse of Dimensionality:** How DNNs succeed in high dimensions
  - Compositional structure, low-dimensional manifolds

Σ: Computational Mathematics and AI Overview



Questions or Feedback?

slido.com
#CBMS25

Concepts

- ML = approximation theory + data
- Bidirectional exchange: CompMath ↔ AI
- Five learning paradigms
- Bias-variance vs. double descent
- Regularization (explicit & implicit)

Insights

- Overparameterization ≠ overfitting
- Minimum norm = implicit regularization
- Classical intuition needs updating
- No Free Lunch
- Domain knowledge/AI literacy matters

References I

- Belkin, M. (2021). “Fit Without Fear: Remarkable Mathematical Phenomena of Deep Learning Through the Prism of Interpolation”. In: *Acta Numerica* 30, pp. 203–248.
- Ferguson, A. et al. (2025). *The Future of Artificial Intelligence and the Mathematical and Physical Sciences (AI+MPS)*. arXiv: 2509.02661 [cs.AI]. URL: https://arxiv.org/abs/2509.02661.
- Higham, C. F. and D. J. Higham (2019). “Deep Learning: An Introduction for Applied Mathematicians”. In: *SIAM Review* 61.4, pp. 860–891.
- Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill.
- Wolpert, D. H. and W. G. Macready (1997). “No Free Lunch Theorems for Optimization”. In: *IEEE Transactions on Evolutionary Computation* 1.1, pp. 67–82.